AD-A105 932 PRINCETON UNIV NJ DEPT OF STATISTICS F/G 12/1
SOME COMPUTATIONAL DETAILS OF CONFIGURAL SAMPLING METHODS.(U)
MAR 81 K BELL, D PREGIBON DAAG29-79-C-0205
UNCLASSIFIED TR-191-SER-2 ARO-16669.10-M NL

1 OF 1
AD A
105932

END
DATE
FILMED
11-81
DTIC

A

5 9

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

LEVEL II

12

AD A105932

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 16669.10-M (19) (18) ARO	2. GOVT ACCESSION NO.: AD-A105 932
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): (6) Some Computational Details of Configural Sampling Methods	5. TYPE OF REPORT & PERIOD COVERED: (9) Technical rept.
	6. PERFORMING ORG. REPORT NUMBER: (14) TR-191-SER-2
7. AUTHOR(s): (10) Katherine Bell, Daryl Pregibon	8. CONTRACT OR GRANT NUMBER(s): (15) DAAG29-79-C-0205
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Princeton University, Princeton, NJ 08544	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Research Office, Post Office Box 12211, Research Triangle Park, NC 27709	12. REPORT DATE: (11) 30 Mar 81
	13. NUMBER OF PAGES: 35 (12) 39
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

DTIC ELECTE OCT 21 1981 S D

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

NA

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Numerical evaluation of the optimum estimate via configural sampling involves evaluation of several double integrals. These integrals represent expectations over a distribution conditioned on the observed configuration. Theoretically, any location-scale invariant definition of the configuration will suffice, though numerically, some choices are better than others. A related concern is the change-of-variables used to map the region of integration, originally the half-plane, onto a fixed region, such as the unit square. This report is of use to the reader both as a guide to the pitfalls and curiosities of the computations presently recommended.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

406873

81 10 20 081

DISCLAIMER NOTICE

Some Computational Details of Configural Sampling Methods*

by

Katherine Bell and Daryl Pregibon

Technical Report No. 191, Series 2
Department of Statistics
Princeton University
1981

*Prepared in connection with research at Princeton University, supported by the Army Research Office (Durham).

ABSTRACT

Numerical evaluation of the optimum estimate via configural sampling involves evaluation of several double integrals. These integrals represent expectations over a distribution conditioned on the observed configuration. Theoretically, any location-scale invariant definition of the configuration will suffice, though numerically, some choices are better than others. A related concern is the change-of-variables used to map the region of integration, originally the half-plane, onto a fixed region, such as the unit square. This report is of use to the reader both as a guide to the pitfalls and curiosities of the computations presently recommended, and as an addendum to Technical Reports 185 and 190 [see references] on the configural polysampling approach.

1. Introduction.

As discussed in Technical Report 187 (Pregibon and Tukey, 1981), configural polysampling techniques are useful in: (1) determining the maximum attainable efficiency in a particular sampling situation, (2) determining the maximum attainable polyefficiency in a particular polysituation, and (3) guiding the modification of a robust estimate with the aim of increasing its polyefficiency. In section 2, we describe the procedure and computations involved in (1) above. Section 3 discusses those involved in (2). Item (3) requires assessing the behavior of an estimate at particular data configurations and will not be discussed in this report. An appendix lists the programs, including the FORTRAN integrator, used in the computations.

2. Single situations.

* background *

Consider a sample $\{x_i : i=1,\ldots,n\}$ from a particular situation $\{f_i : i=1,\ldots,n\}$ where the f_i are location-scale densities. Following Bruce, Pregibon and Tukey (1981), the situation is termed simple if $f_i = f$ for all i; otherwise the situation is termed compound. For example

*Prepared in connection with research at Princeton University, supported by the Army Research Office (Durham).

$$X_i \sim \text{Gau}(0,1) \qquad i=1\ldots,n$$

is a simple situation whereas

$$n-1 \; X\text{'s} \sim \text{Gau}(0,1)$$

$$\text{one } X \sim \text{Gau}(0,100)$$

is a compound situation.

Configural methods require transformation from the observed sample to its location-scale invariant representation. In most cases, the configuration is expressed via transformation of the order statistics $y_1 \leq y_2 \leq \cdots \leq y_n$. The general form of the change-of-variables is

$$r = r(\underline{y})$$

$$s = s(\underline{y})$$

$$c_i = (y_i - r)/s \qquad i=1,\ldots,n \; ,$$

where r is a measure of location and s a measure of scale.

Configural methods restrict attention to location-scale invariant estimators $t(\underline{y}) = t(r+s\underline{c}) = r+st(\underline{c})$. This allows the determination of the minimum mean squared error (MSE) estimate of location conditional on the observed configuration $\{c_i : i=1,\ldots,n\}$. Without loss of generality, assume that f_i is centered at $\mu=0$ with scale $\sigma=1$. Then the conditional mean squared error of the estimate is

$$MSE\{t(\underline{y})|\underline{c}\} = E_{r,s}\{r+st(\underline{c})|\underline{c}\}^2 .$$

This quantity is minimized by

$$t_o(\underline{c}) = -E\{rs|\underline{c}\}/E\{s^2|\underline{c}\}$$

with

$$MSE\{t_o(\underline{y})|\underline{c}\} = E\{rs|\underline{c}\}\ t_o(\underline{c}) + E\{r^2|\underline{c}\} .$$

Averaging $MSE(t_o(y)|\underline{c})$ over the distribution of configurations provides an estimate of the unconditional variance at $t_o(\underline{y})$. The estimate $t_o(\underline{y})$ is unconditionally minimum variance for a symmetric situation since in that case the unconditional bias is zero. For any particular sample, the optimal estimate and its conditional MSE can be computed by numerical evaluation of the conditional expectations as we now describe.

* computational details *

Samples are generated in a subroutine, and passed in common to the main program. The program is shown in listing 1 in the appendix. The data may correspond to a sample from either a simple or compound situation. A sorting subroutine (listing 4 in the appendix) provides the order statistics $y_1 \leq \cdots \leq y_n$.

The configuration $\{c_i\}$ is formed by making the change of variables

$$r = r(\underline{y})$$

$$s = s(\underline{y})$$

$$c_i = (y_i - r)/s \qquad i=1,\ldots,n \ .$$

The Jacobian of this transformation is s^{n-2}. Thus, in terms of our new coordinates, we have the probability element

$$f(\underline{y})d\underline{y} = s^{n-2} f(r+s\underline{c})\,dr\,ds\,d\underline{c} \ .$$

The marginal density of $\underline{c}$ is

$$h(\underline{c}) = \int_r \int_s s^{n-2} f(r+s\underline{c})\,dr\,ds \ .$$

The range of integration in this expression is the half-plane. In order to improve the accuracy of a fixed-point quadrature, we map the half-plane onto the unit-square via (see Relles and Rogers, 1977):

$$u = 1/(1+\exp[n^{\frac{1}{2}}(\log s - \log s^*)]) \qquad 0 \le u \le 1$$

$$v = 1/(1+\exp[n^{\frac{1}{2}}(r-r^*)/s]) \qquad 0 \le v \le 1$$

where s^* and r^* are appropriate centering values for the bivariate conditional density

$$g(r,s|\underline{c}) = s^{n-2} f(r+s\underline{c})/h(\underline{c}) \ .$$

The Jacobian of this transformation is

$$J(u,v) = \frac{s^{*2}}{nu^2 v}\left(\frac{1-u}{u}\right)^{\frac{2}{\sqrt{n}}-1}\left(\frac{1}{1-v}\right) .$$

Thus, in terms of our new coordinates, we have the probability element

$$f(\underline{y})dy = J(u,v)s(u,v)^{n-2}f(r(u,v)+s(u,v)\underline{c})dudvd\underline{c}$$

$$= g(u,v,\underline{c})dudvd\underline{c} .$$

* cubature *

The evaluation of the required conditional expectations can now be carried out by two-dimensional numerical integration (cubature). The following integrals (each defined on the unit square):

(1) $h(\underline{c}) = \iint g(u,v,\underline{c})dudv$

(2) $E(s^2|\underline{c})h(\underline{c}) = \iint s(u,v)^2 g(u,v,\underline{c})dudv$

(3) $E(r^2|\underline{c})h(\underline{c}) = \iint r(u,v)^2 g(u,v,\underline{c})dudv$

(4) $E(rs|\underline{c})h(\underline{c}) = \iint r(u,v)s(u,v)g(u,v,\underline{c})dudv.$

have so far been done using a 24 point Gaussian quadrature rule in both dimensions (but see below). Thus, for example, (1) is computed as

$$h(\underline{c}) = \sum_{j=1}^{24}\sum_{k=1}^{24} w_j w_k \; g(z_j,z_k,\underline{c})$$

where $\{w_i : i=1,...,24\}$ and $\{z_i : i=1,...,24\}$ are optimally

chosen weights and evaluation points along one dimension. In particular, these values are chosen so that the finite sum is exactly $h(\underline{c})$ for one dimensional polynomials $g(z)$ up to degree 47 ((Krylov, 1962, pp.110-111 and 337-340), (Abramowitz and Stegun, 1970)). The two dimensional integrator is exact for a function $g(z1,z2)$ such that $g(z2|z1)$, the function conditioned on the value of z1, is a 47 degree polynomial and such that the one dimensional integrals are a 47 degree polynomial. The two dimensional integrator is thus exact for a function $g(z1,z2)$ which is not above degree 47 in either of the two variables.

A listing of the one dimensional Gaussian quadrature subroutine used in the calculations is given in the Appendix (listing 5). Figure 1 shows the grid of points (z_j,z_k) on the unit square at which the bivariate function is evaluated in the integration. Figure 2 shows the grid of quadrature coefficients, $w_j w_k$, used in the 24 point quadrature. The values shown are the quadrature coefficients for the 144 points in the quarter-square $(0<z_j<.5,\ 0<z_k<.5)$, where each weight has been multiplied by 10^5. Note that the weights have been plotted on an equally spaced grid but that the weights shown in figure 2 are associated with points on the unequally spaced grid (figure 1). (The numbers of the points (1-24) with which the weights are associated are labelled in the figure.) The coefficients for the 576 points on the unit square are derived from the values shown in figure 2 using the fact that

quadrature coefficient (.5+c) = quadrature coeffieient (.5-c)

for the values of c used in the quadrature program. Figure 3 shows the values of $\log_{10}(w_j w_k)+6$. These are again plotted on an equally spaced grid but are associated with the points of figure 1.

As noted, the 24 point Gaussian quadrature integrates polynomials up to 47 exactly, and was useful for testing purposes. We anticipate reducing the number of points evaluated to fewer than 576 for post-testing computations.

The two dimensional integration is obtained by providing the integrator a function which is itself a one-dimensional integral. In essence, the subroutine calls itself. However, since recursive function calls are not supported in Fortran, the subroutine must invoke a copy of itself compiled under a different name. The function argument of the call to the copy of the integrator does the actual functional evaluations $g(z_j,z_k,\underline{c})$. As each of the integrals (1)-(4) has kernel $g(u,v,\underline{c})$, the 24x24 grid of values of $g(z_i,z_k,\underline{c})$ need only be computed once. We take advantage of this property by storing the matrix $g(z_i,z_k,\underline{c})$ after evaluation of (1), and using these values for evaluation of (2) - (4). This provides us with the quantities

$h(\underline{c}) = (1)$

$E(s^2|\underline{c}) = (2)/(1)$

$E(r^2|\underline{c}) = (3)/(1)$

$E(rs|\underline{c}) = (4)/(1)$

as are needed in calculating $t_o(\underline{c})$ and $MSE(t_o(\underline{y})|\underline{c})$.

The output from a typical run of the program is a $(N+1) \times 7$ array of the form:

$h(\underline{c}_1)$	$E(s^2|\underline{c}_1)$	$E(rs|\underline{c}_1)$	$E(r^2|\underline{c}_1)$	$t_o(\underline{c}_1)$	$t_o(\underline{y}_1)$	$MSE(t(\underline{y}_1|\underline{c}_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$h(\underline{c}_N)$	$E(s^2|\underline{c}_N)$	$E(rs|\underline{c}_N)$	$E(r^2|\underline{c}_N)$	$t_o(\underline{c}_N)$	$t_o(\underline{y}_N)$	$MSE(t(\underline{y}_N)|\underline{c}_N)$
$h(\underline{c})$	$E(s^2)$	$E(rs)$	$E(r^2)$	$t_o(\underline{c})$	$t_o(\underline{y})$	$MSE(t(\underline{y}))$

Each of the first N rows, corresponds to estimates of the conditional expectations given an individual configuration. The final row provides the estimates of the unconditional expectation obtained as the average over configurations.

* major choices *

There are several choices in the computational procedure outlined above which have an effect on the accuracy of the results. These include the choice of r^* and s^* and the forms of $r(\underline{y})$ and $s(\underline{y})$ used in the transformation from the data to the configuration. We now discuss these.

Relles and Rogers (1977) use the transformation $(r,s) \to (u,v)$ where r^* and s^* are the points at which the density

$$s^{n-2} \cdot f(r+s\underline{c})$$

attains its maximum. They state that this transformation causes the functions we integrate to be more closely constant on their domains. To reduce computation time and expense, it appears advantageous to choose r* and s* by a method other than that suggested by Relles and Rogers.

The possibility of using

$$r^* = r_{obs}$$

$$s^* = s_{obs}$$

has been tested for various functional forms of r and s. The form

$$r = y(1)$$

$$s = y(n)-y(1) \quad ,$$

i.e., the minimum as the location estimate and the range of the data as the scale estimate has the property of putting the configuration on the interval [0,1]. However, when these forms are used with $r^*=r_{obs}$ and $s^*=s_{obs}$, the estimates produced for some samples are very inaccurate.

The problem with this approach can be seen in a close look at the integration for a straggling sample. Samples with large values of y(n)-y(1) were observed when the data were generated from the slash. The density of the slash is

$$\frac{1}{\sqrt{2\pi}\,y^2}\left(1-\exp\{-\tfrac{1}{2}y^2\}\right) \qquad \text{for } y\neq 0$$

$$\frac{1}{2\sqrt{2\pi}} \qquad \text{for } y=0\ .$$

Alternately, slash is defined as the ratio of an independent Gaussian to a uniform (0,1) random variable. The slash density is like the Gaussian in the middle and like the Cauchy in the tails, and so has much longer tails than the Gaussian.

In samples with a large range the contribution from several points on the 24x24 grid used in the quadrature swamp all others and the double integration reduces to the weighted sum of the values of the function at only a few points. Figure (4) shows the 24x24 grid of powers of 10^{-1} of the values of $g(u,v,\underline{c})$ used, for a particular configuration, in evaluating the double integral. This plot is for a sample of n=20 with y(20)-y(1) = 41,722, and with y(15)-y(5) = 2.808. The values here and in the figures 5-8 are shown on an equally spaced grid, but correspond to points on the grid shown in figure 1.

As an alternative, the location and scale measures

r = midpivot = mean of the pivots

s = pivotspread = difference of the pivots

were tried and used with $r^*=r_{obs}$ and $s^*=s_{obs}$ in the second transformation. The pivot depth is defined as the integer

part of the hinge depth, i.e., for sample size n, pivot depth = $\left[\frac{1}{2}\left[\frac{n+1}{2}\right]+\frac{1}{2}\right]$ where the brackets denote integer part. The pivots are then the order statistics with depth = pivot depth and the midpivot is the average of the two pivots. Figure (5) shows the 24x24 grid of powers of 10^{-1} of the values of the function $g(u,v,\underline{c})/J(u,v)$ (i.e., without the Jacobian $J(u,v)$ from the second transformation) when these new values are used. Figure (6) is the comparable plot for the function $g(u,v,\underline{c})$. Comparing figures (4) and (5), we see a much more constant order of magnitude of the function over the domain when the midpivot and pivotspread are used. Figure (7) shows the grid of powers of 10^{-1} of the values of the product of the function $g(u,v,\underline{c})$ and the quadrature coefficients used in the integration.

In an attempt to make the surface we integrate over still more constant, r* and s* were moved to correspond to (*) in figure (5). The results are shown in figure (8), where the values plotted on the grid are again powers of 10^{-1} for the function values. Also of interest is the change in the optimum estimate for the original

$$r^* = r_{obs} = \text{midpivot}$$

$$s^* = s_{obs} = \text{pivotspread}$$

and the relocated r* and s*. The estimate values are -.7076168 and -.7074012, respectively. This small change (.0002156) in the values of the estimate leads us to ques-

tion the gain from recentering.

3. Bisampling.

In the previous section, computations for the case when data are generated from and used as if they are from the same situation were described. In bisampling (see listing 2 in the appendix), we distinquish between the generating situation and the evaluating situation. The former is the situation actually generating the data, while the latter is the situation we treat the data as being from and at which we evaluate the optimum estimate.

Suppose we have two situations, for example, slash and Gaussian, f_s and f_G. We generate a sample from the Gaussian and proceed as described in the previous section to calculate the minimum variance estimate for the associated configuration. Here the Gaussian is both the generating and the evaluating distribution. We then use the same data and configuration and treat it as being generated by slash, i.e. we have a Gaussian generating and a slash evaluating situation. A similar procedure is followed with generated slash data.

In bisampling, we also calculate weights, w_G and w_s as

$$w_G = f_G(\underline{c})/(\alpha_G \cdot f_G(\underline{c})+\alpha_s \cdot f_s(\underline{c}))$$

$$w_s = f_s(\underline{c})/(\alpha_G \cdot f_G(\underline{c})+\alpha_s \cdot f_s(\underline{c}))$$

where α_G and α_s are the sampling fractions, $N_G/(N_G+N_s)$ and $N_s/(N_G+N_s)$, for the Gaussian and slash, respectively. w_G is

the weight proportional to the probability that the configuration is Gaussian given that the configuration is one of N_G Gaussian configurations or N_S slash configurations; w_S is defined similarly for the slash. These weights are used in calculating the average MSE_G and MSE_S.

The output from this program is

$$w_G \qquad E_G\{s^2|\underline{c}\} \qquad t_o^G(y) \qquad MSE_G$$

$$w_S \qquad E_S\{s^2|\underline{c}\} \qquad t_o^S(\underline{y}) \qquad MSE_S$$

for each of the samples from the Gaussian and for each of the samples from the slash. Evaluation of the maximum attainable biefficiency (for slash and Gaussian data) using this output is presently under consideration (see listing 3 in the appendix and (Tukey, 1981a)).

4. Conclusions.

Computing the optimum estimate for a situation using configural sampling or configural polysampling methods involves the evaluation of several double integrals. The choices of (1) functional forms of r and s used in transforming the data to the configurations, and (2) the values of r* and s* as appropriate central values of the bivariate density $f_{\underline{c}}$ (r,s) affect the precision and accuracy of the numerical integrations. The choices

r = midpivot

s = pivotspread

and

$r^* = r_{obs}$

$s^* = s_{obs}$

give well-balanced functions and, thereby, good integral evaluations and estimates. This choice also keeps computation costs to a reasonable level and below those of some alternative choices.

REFERENCES

Abramowitz, Milton and Stegun, Irene A., 1970. Handbook of Mathematical Functions, Dover Publications, Inc., New York, 916-919.

Krylov, V. I., 1966. Approximate Calculation of Integrals, MacMillan, New York.

Pregibon, D. and Tukey, J. W., 1981. "Assessing the behavior of robust estimates of location in small samples: Introduction to configural polysampling," Technical Report No. 185, Series 2, Department of Statistics, Princeton University.

Relles, D.A. and Rogers, W.H., 1977. "Statisticians are fairly robust estimators of locations," JASA, 72, March, 107-111.

Tukey, J.W., 1981a. "Some advanced thoughts on the data analysis involved in configural polysampling directed toward high performance estimates," Technical Report No. 189, Series 2, Department of Statistics, Princeton University.

Tukey, J.W., 1981b. "Kinds of polyconfidence limits for centers, and some thoughts on identification and selection of confidence procedures using configural polysampling," Technical Report No. 190, Series 2, Department of Statistics, Princeton University.

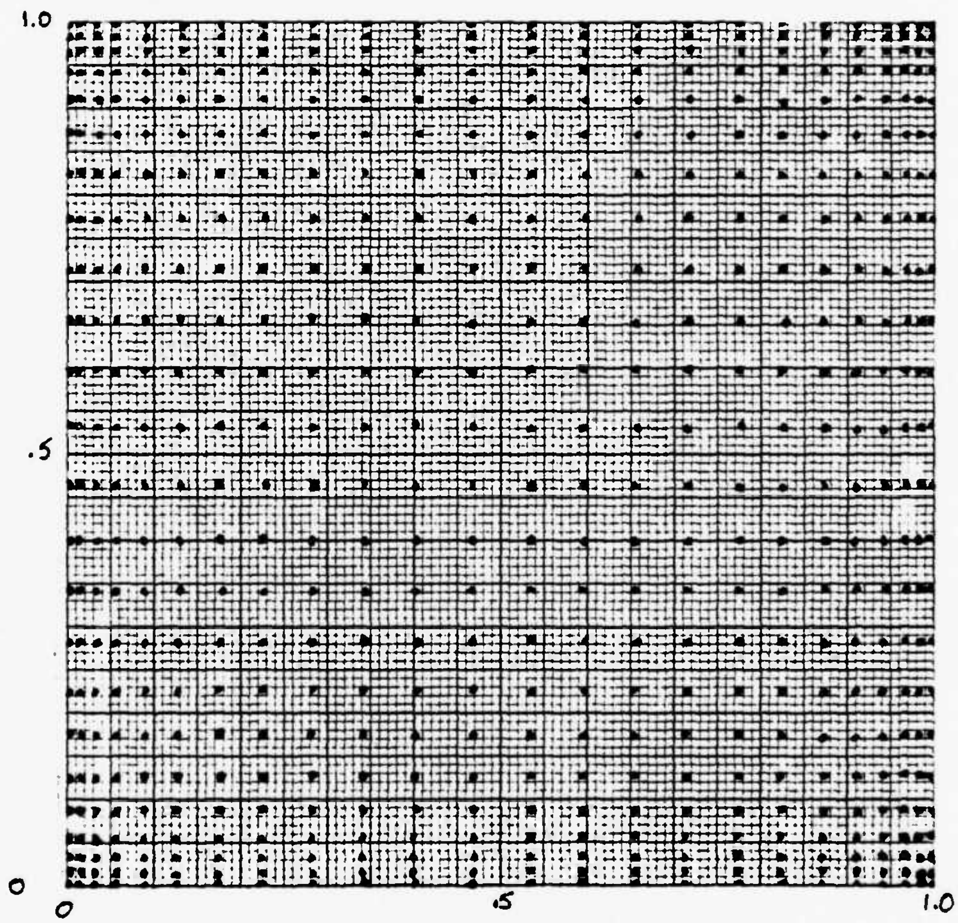

Figure 1: Grid of points (z_j, z_k) at which the bivariate function is evaluated for the 24 point quadrature.

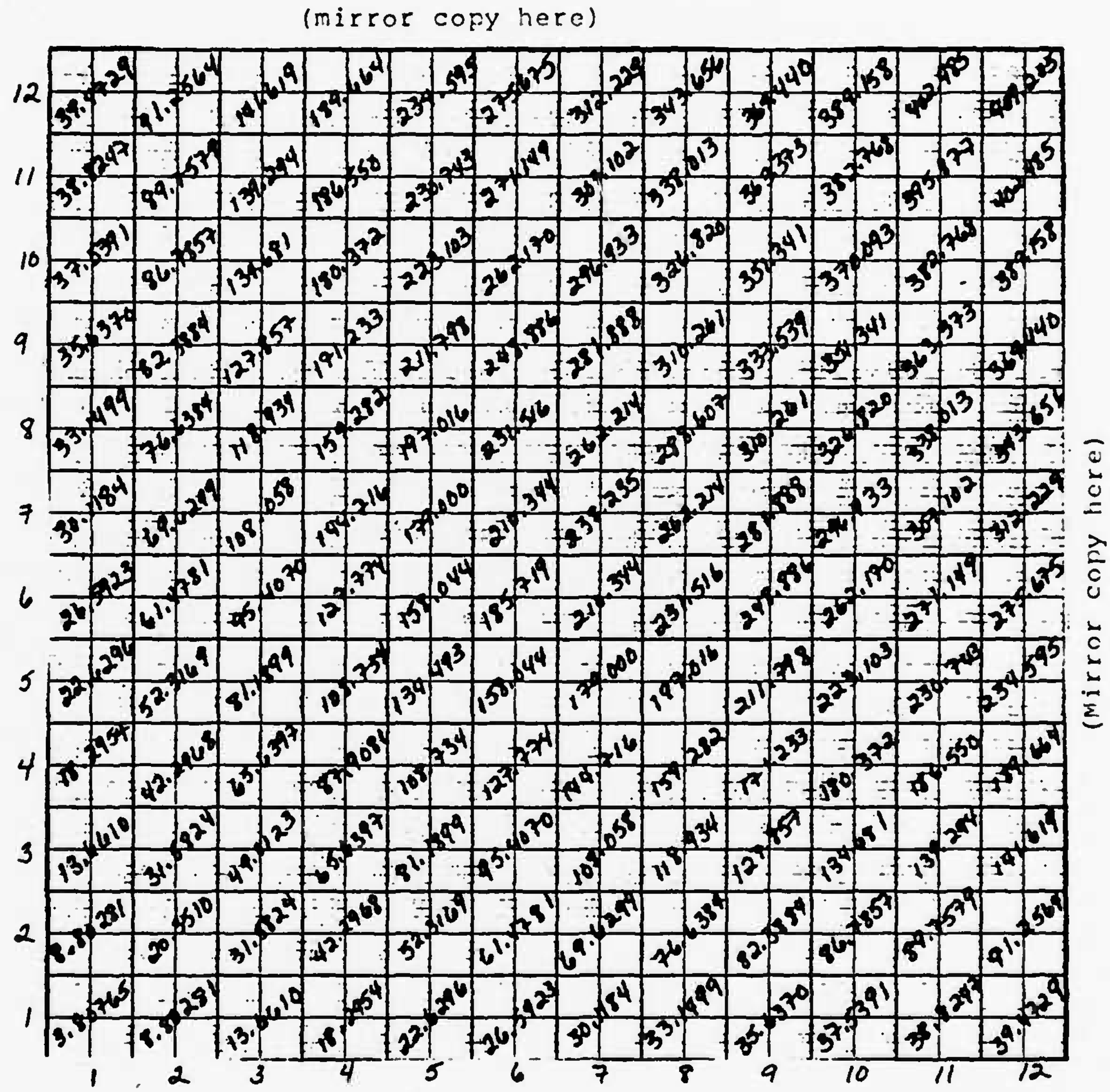

Figure 2: Quadrature coefficients (multiplied by 10^5) for the quarter-square ($0<z_j<.5$; $0<z_k<.5$).

(mirror copy here)

12	2.5963	2.9603	3.1511	3.2780	3.3703	3.4404	3.4945	3.5361	3.5675	3.5901	3.6047	3.6119
11	2.5891	2.9531	3.1439	3.2708	3.3631	3.4332	3.4873	3.5289	3.5604	3.5829	3.5936	3.6047
10	2.5745	2.9354	3.1293	3.2562	3.3485	3.4186	3.4727	3.5143	3.5457	3.5683	3.5829	3.5901
9	2.5519	2.9159	3.1067	3.2336	3.3259	3.3960	3.4501	3.4917	3.5231	3.5457	3.5604	3.5675
8	2.5205	2.8844	3.0753	3.2022	3.2945	3.3646	3.4187	3.4603	3.4917	3.5143	3.5289	3.5361
7	2.4788	2.8428	3.0337	3.1605	3.2529	3.3229	3.3770	3.4187	3.4501	3.4727	3.4873	3.4945
6	2.4248	2.7887	2.9796	3.1064	3.1988	3.2689	3.3229	3.3646	3.3960	3.4186	3.4332	3.4424
5	2.3547	2.7186	2.9095	3.0364	3.1287	3.1988	3.2529	3.2945	3.3259	3.3485	3.3631	3.3703
4	2.2623	2.6263	2.8172	2.9440	3.0364	3.1064	3.1605	3.2022	3.2336	3.2562	3.2708	3.2780
3	2.1355	2.4994	2.6903	2.8172	2.9095	2.9796	3.0337	3.0753	3.1067	3.1293	3.1439	3.1511
2	1.9446	2.3086	2.4994	2.6263	2.7186	2.7887	2.8428	2.8844	2.9159	2.9354	2.9531	2.9603
1	1.5807	1.9446	2.1355	2.2623	2.3547	2.4248	2.4788	2.5205	2.5519	2.6745	2.5891	2.5963
	1	2	3	4	5	6	7	8	9	10	11	12

(mirror copy here)

Figure 3: $\log_{10}$ (quadrature coefficient) +6
for the quarter-square $(0<z_j<.5;\ 0<z_k<.5)$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
24	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
23	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
22	43	67	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
21	65	14	58	66	71	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
20	73	61	38	47	60	66	70	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
19	73	72	61	44	32	53	61	66	70	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73
18	73	63	70	62	49	-19	46	57	63	67	71	73	73	73	73	73	73	73	73	73	73	73	73	73
17	73	73	73	69	62	52	21	38	53	60	65	69	72	73	73	73	73	73	73	73	73	73	73	73
16	73	73	73	73	69	63	54	57	29	50	68	69	69	71	73	73	73	73	73	73	73	73	73	73
15	73	73	73	73	73	69	63	55	41	17	47	67	63	68	71	73	73	73	73	73	73	73	73	73
14	73	73	73	73	73	73	68	63	56	43	6	45	56	63	68	71	73	73	73	73	73	73	73	73
13	73	73	73	73	73	73	72	68	63	56	45	-37	45	56	63	68	72	73	73	73	73	73	73	73
12	73	73	73	73	73	73	73	72	68	63	56	45	-38	45	57	64	68	73	73	73	73	73	73	73
11	73	73	73	73	73	73	73	73	72	68	63	56	45	-1	47	58	65	70	73	73	73	73	73	73
10	73	73	73	73	73	73	73	73	73	72	68	63	56	43	16	49	60	67	72	73	73	73	73	73
9	73	73	73	73	73	73	73	73	73	73	72	68	63	56	41	28	53	63	69	73	73	73	73	73
8	73	73	73	73	73	73	73	73	73	73	73	72	68	63	55	38	37	56	66	72	73	73	73	73
7	73	73	73	73	73	73	73	73	73	73	73	73	72	68	63	54	32	49	60	68	73	73	73	73
6	73	73	73	73	73	73	73	73	73	73	73	73	73	72	68	63	53	17	50	69	73	73	73	73
5	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	69	62	51	15	56	69	73	73	73
4	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	69	63	49	37	62	73	73	73
3	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	71	63	47	48	68	73	73
2	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	65	47	56	73	73
1	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	73	71	54	61	73

Figure 4: Powers of 10^{-1} for the values of $g(u,v,\underline{c})$ at which the function is evaluated for integration ($r=y(1)$; $s=y(n)-y(1)$). (Note values of -37 and -38; these are 10^{17} times the next largest value).

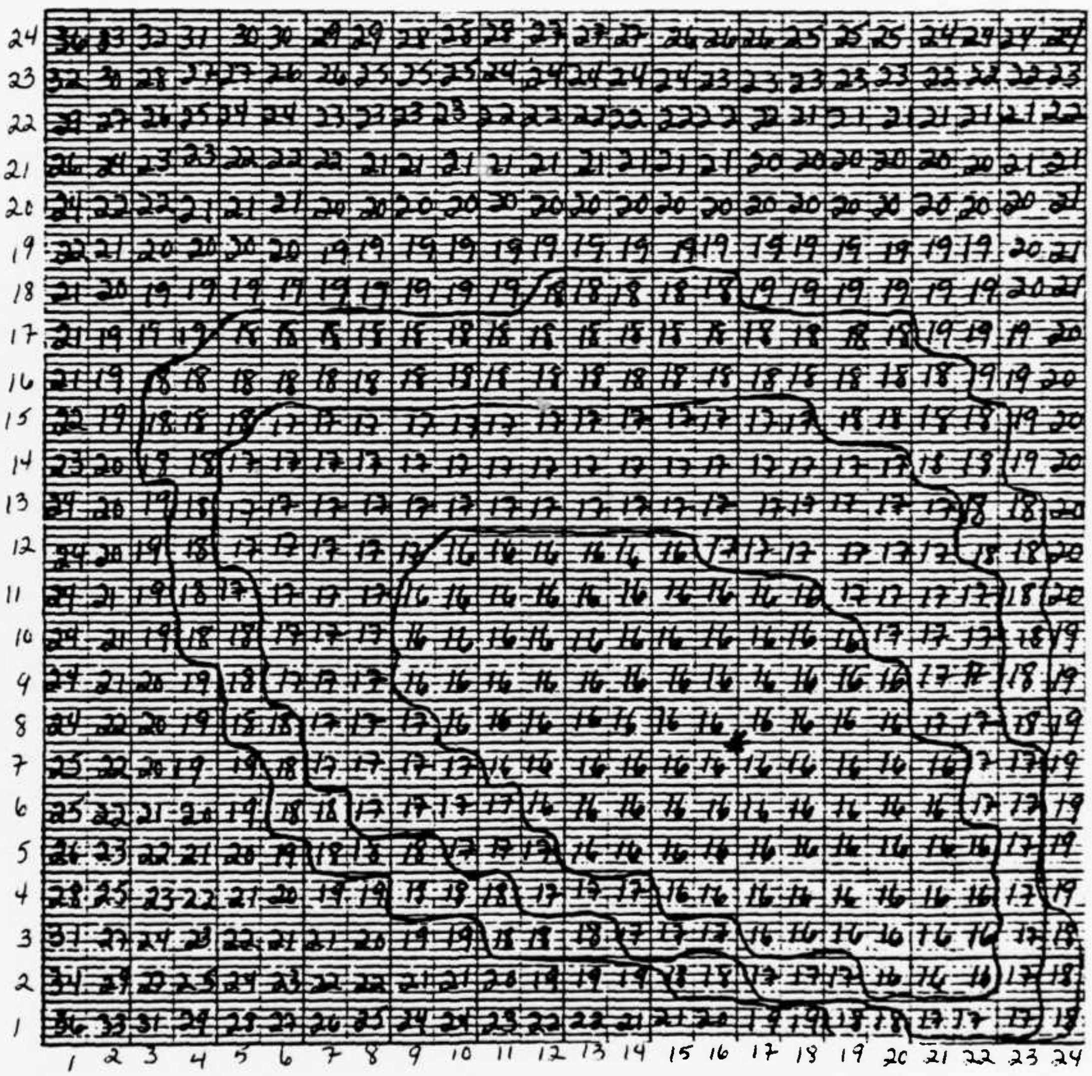

Figure 5: Powers of 10^{-1} for the values of $g(u,v,\underline{c})/J(u,v)$ at which the function is evaluated for integration (r=midpivot; s=pivotspread).

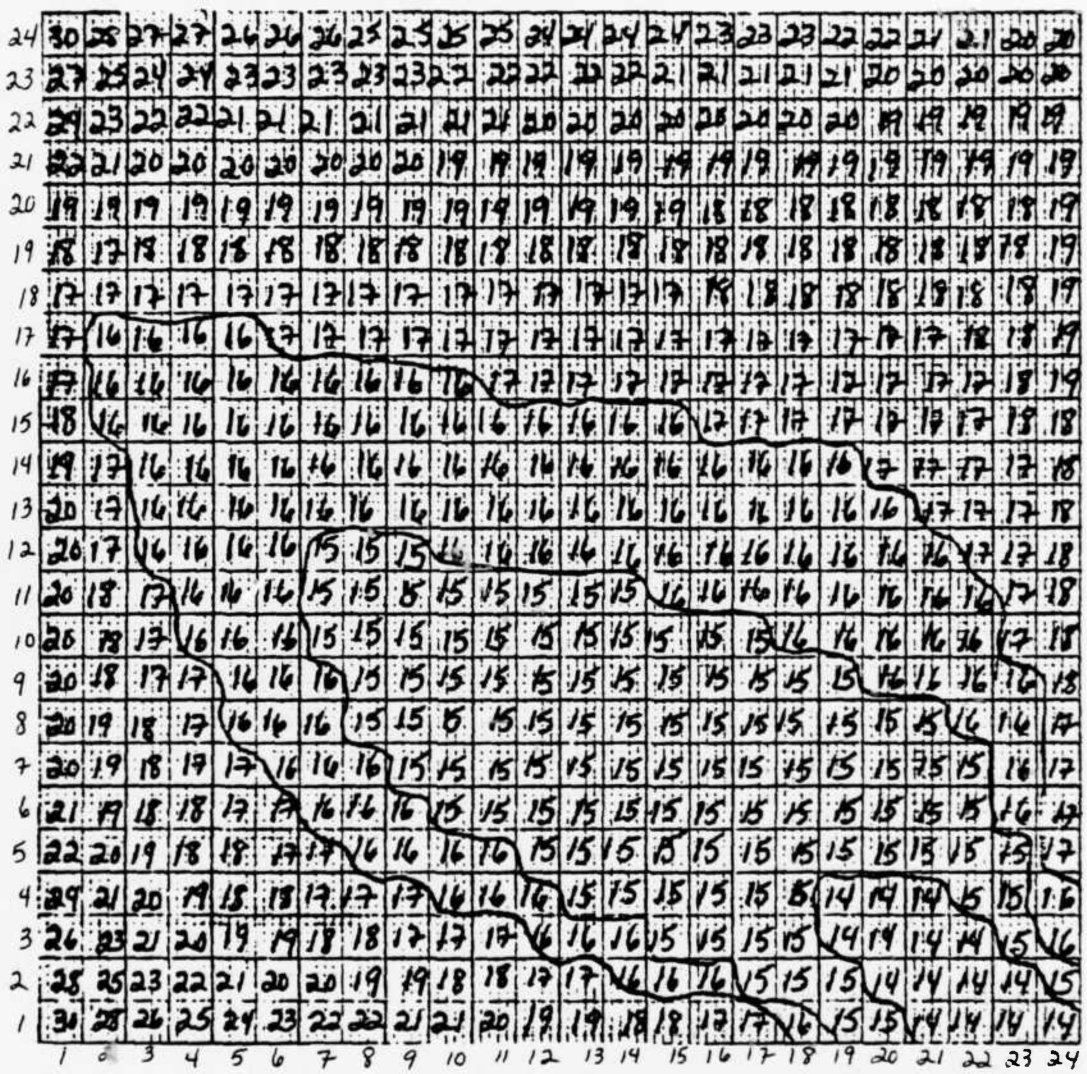

Figure 6: Powers of 10^{-1} for the values of $g(u,v,\underline{c})$ at which the function is evaluated for integration (r=midpivot; s=pivotspread).

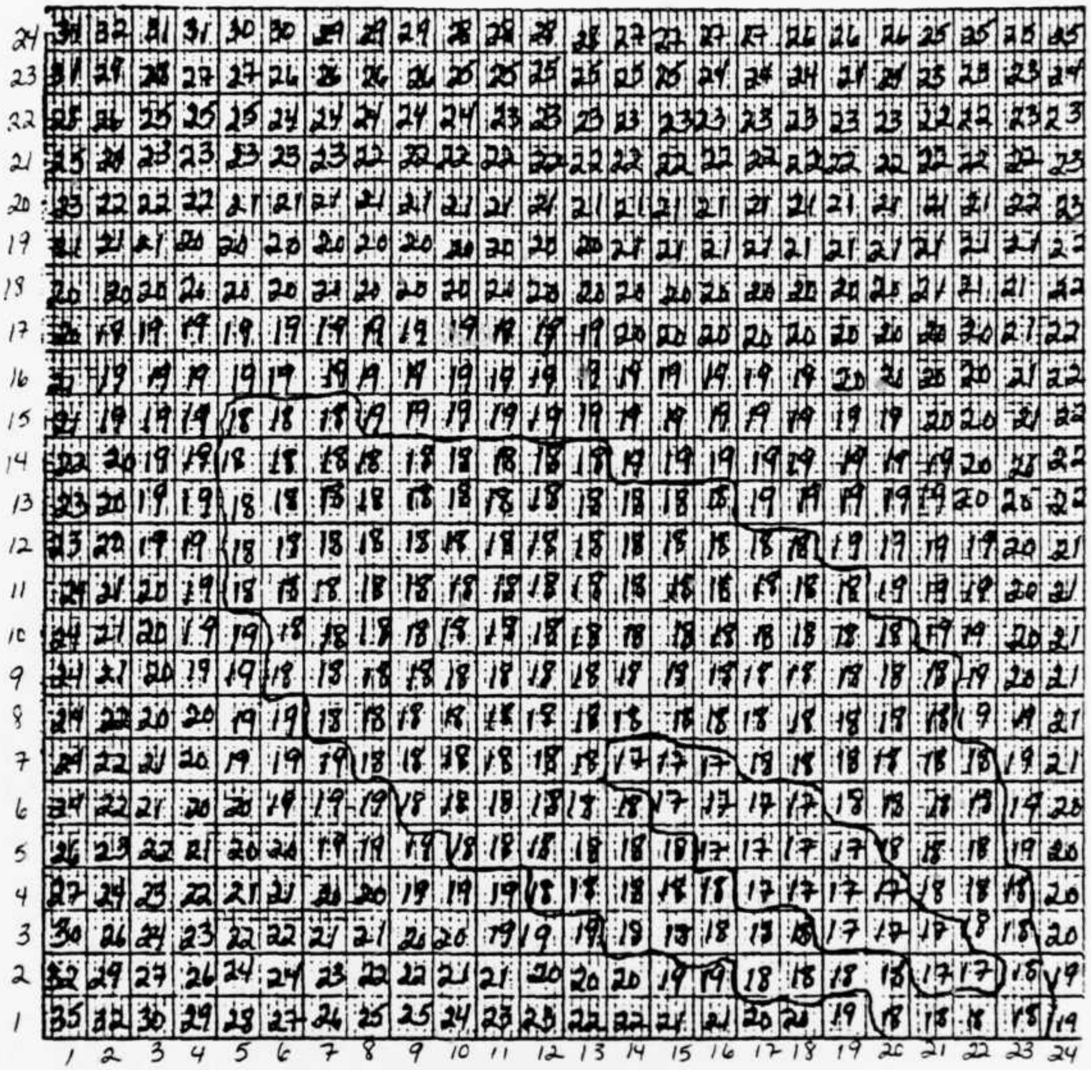

Figure 7: Powers of 10^{-1} for the values of the product $g(u,v,\underline{c})\cdot$(quadrature coefficient) at which the function is evaluated for integration (r=midpivot; s=pivotspread).

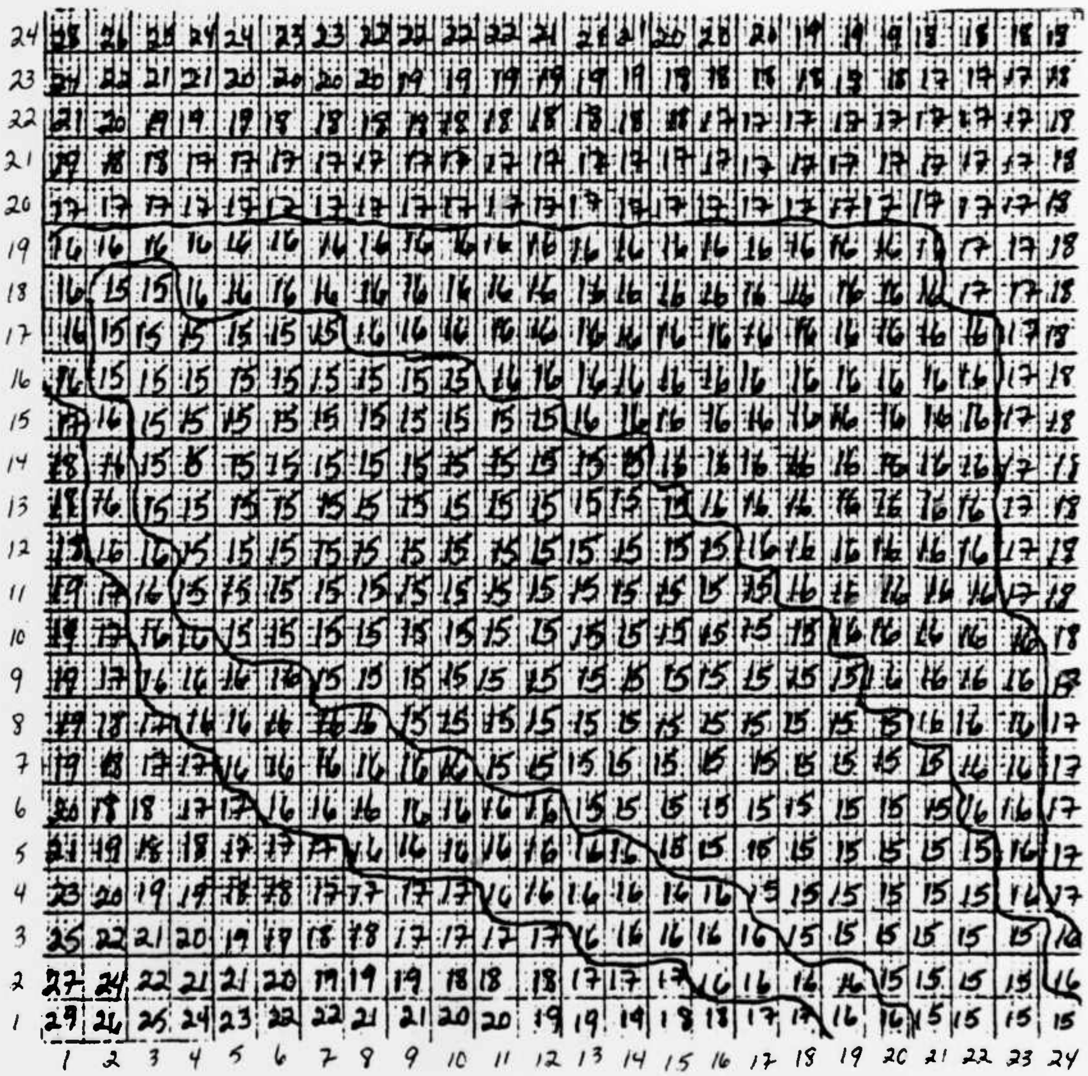

Figure 8: Powers of 10^{-1} for the values of $g(u,v,\underline{c})$ at which the function is evaluated for integration (r^* and s^* relocated).

Appendix

```
C        LISTING 1
C        COMPUTES EXPECTED VALUES, T(C),T(Y), AND MSE(T(Y))
C        FOR ONE GENERATING DISTRIBUTION AND ONE EVALUATING
C        DISTRIBUTION
C
         IMPLICIT REAL*8 (A-H,O-Z)
         REAL*8 C(100),AVE(7)
         COMMON /AREA1/C,N/AREA4/PST,SST/AREA2/L,K,J/AREA3/V/AREA5/IND
         COMMON /AREA6/CONT,ADJ
         EXTERNAL FINT,FINT0
         DSEED=13271311.D0
C        READ SAMPLE SIZE, NUMBER OF SAMPLES, EVALUATING AND
C        GENERATING DISTRIBUTIONS, CONTAMINATION FOR ONG
C        FOR L AND LL, 1:GAUSSIAN, 2:ONG, 3:SLASH, 4:LOGISTIC,
C        5:CAUCHY.
         READ(5,150) N,NSAMP,L,LL,CONT
 150     FORMAT(1I2,1I3,2I2,F3.0)
         IF(N .EQ. 0) GO TO 600
         ADJ=1.D0-1.D0/CONT
         CONT=DSQRT(CONT)
         DNSAMP=DFLOAT(NSAMP)
         DO 200 II=1,7
 200     AVE(II)=0.D0
         CONS=CONST(L)
         DO 400 LOOP=1,NSAMP
C        GENERATE AND SORT DATA
         CALL RANDEV(LL,DSEED)
         CALL SORT8(C,N)
         WRITE(8,300) (C(I),I=1,N)
         NN=(N+1)/2
         NN1=(NN+1)/2
         NN2=N-NN1+1
C        P=MIDPIVOT AND S=PIVOTSPREAD
         P=(C(NN1)+C(NN2))/2.D0
         S=C(NN2)-C(NN1)
C        TRANSFORM DATA TO CONFIGURATION
         DO 250 I=1,N
 250     C(I)=(C(I)-P)/S
         PST=P
         SST=S
C        EVALUATION OF DOUBLE INTEGRALS
         K=0
         J=0
         IND=0
         CALL DINT24(FINT,F00)
         K=2
         J=0
         IND=0
         CALL DINT24(FINT0,F20)
         F20=F20/F00
         K=1
         J=1
         IND=0
```

```
      CALL DINT24(FINT0,F11)
      F11=F11/F00
      K=0
      J=2
      IND=0
      CALL DINT24(FINT0,F02)
      F02=F02/F00
      F00=F00*CONS
      PITC=-F11/F20
      PITX=R+S*PITC
      PITMSE=F11*PITC+F02
C     WRITE OUT NEEDED DOUBLE INTEGRALS, T(C),T(X),MSE(T(X))
      WRITE(8,300)F00,F20,F11,F02,PITC,PITX,PITMSE
 300  FORMAT(7D16.7)
      AVE(1)=AVE(1)+F00
      AVE(2)=AVE(2)+F20
      AVE(3)=AVE(3)+F11
      AVE(4)=AVE(4)+F02
      AVE(5)=AVE(5)+PITC
      AVE(6)=AVE(6)+PITX
      AVE(7)=AVE(7)+PITMSE
 400  CONTINUE
      DO 500 II=1,7
 500  AVE(II)=AVE(II)/DNSAMP
C     WRITE OUT AVERAGE VALUES OF QUANTITIES WRITTEN OUT ABOVE
      WRITE(8,550)
 550  FORMAT(' AVERAGES OVER CONFIGURATIONS ARE:')
      WRITE(8,300)(AVE(II),II=1,7)
 600  CONTINUE
      STOP
      END
C
      DOUBLE PRECISION FUNCTION FINT(V)
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL GINT,GINT0
      COMMON /AREA3/VV
      VV=V
      CALL AINT24(GINT,ANS)
      FINT=ANS
      RETURN
      ENTRY FINT0(V)
      VV=V
      CALL AINT24(GINT0,ANS)
      FINT0=ANS
      RETURN
      END
C
      SUBROUTINE RANDEV(LL,DSEED)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 C(100)
      COMMON /AREA1/C,N/AREA6/CONT,ADJ
      DATA PI/3.14159265357828/
      GO TO (10,20,30,40,50),LL
```

```
10    DO 15 I=1,N
15    C(I)=GGNQF(DSEED)
      RETURN
20    DO 25 I=1,N
25    C(I)=GGNQF(DSEED)
      C(N)=CONT*C(N)
      RETURN
30    DO 35 I=1,N
35    C(I)=GGNQF(DSEED)/GGUBFS(DSEED)
      RETURN
40    DO 45 I=1,N
      ARG=GGUBFS(DSEED)
45    C(I)=DLOG(ARG/(1.D0-ARG))
      RETURN
50    DO 55 I=1,N
55    C(I)=DTAN(PI*(GGUBFS(DSEED)-0.5D0))
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION CONST(L)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 C(100)
      COMMON /AREA1/C,N
      DATA PI/3.14159265357828/
      CONST=1.D0
      GO TO (1,1,1,3,2),L
1     CONST=1.D0/DSQRT(2.D0*PI)
      CONST=CONST**N
      RETURN
2     CONST=(1.D0/PI)**N
3     RETURN
      END
C
      DOUBLE PRECISION FUNCTION GINT(U)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 Z(100),TEMPS(576),TEMPR(576),TEMP(576)
      COMMON/AREA1/Z,N/AREA4/PST,SST/AREA2/L,K,J/AREA3/V/AREA5/IND
      COMMON /AREA6/CONT,ADJ
      IND=IND+1
      DN=DFLOAT(N)
      POW=1.D0/DSQRT(DN)
      TEMPS(IND)=SST*(((1.D0-U)/U)**POW)
      TEMPR(IND)=TEMPS(IND)*POW*DLOG((1.D0-V)/V)+PST
      TEMP(IND)=DN*DLOG(TEMPS(IND))-DLOG(DN*U*V*(1.D0-U)*(1.D0-V))
      SUM1=0.D0
      SUM2=0.D0
      GO TO(10,20,30,40,50),L
10    DO 15 I=1,N
      X=TEMPR(IND)+TEMPS(IND)*Z(I)
15    SUM1=SUM1-X*X/2.D0
      GO TO 60
20    DO 25 I=1,N
      X=TEMPR(IND)+TEMPS(IND)*Z(I)
```

```
      X2D2=X*X/2.D0
      EXPON=X2D2*ADJ
      IF(EXPON .GT. 170.D0) EXPON=170.D0
      SUM1=SUM1-X2D2
25    SUM2=SUM2+DEXP(EXPON)/(DN*CONT)
      SUM1=SUM1+DLOG(SUM2)
      GO TO 60
30    DO 35 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
      X2=X*X
      EXPON=0.5D0*X2
      IF(EXPON .GT. 170.D0) EXPON=170.D0
      SUM2=1.D0-DEXP(-EXPON)
35    SUM1=SUM1+DLOG(SUM2/X2)
      GO TO 60
40    DO 45 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
      EXPON=X
      IF(DABS(EXPON) .GT. 170.D0) EXPON=DSIGN(170.D0,EXPON)
      SUM2=1.D0+DEXP(EXPON)
45    SUM1=SUM1+X-2.D0*DLOG(SUM2)
      GO TO 60
50    DO 55 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
55    SUM1=SUM1-DLOG(1.D0+X*X)
60    TMP=TEMP(IND)+SUM1
      IF(DABS(TMP) .GT. 170.D0) TMP=DSIGN(170.D0,TMP)
      TEMP(IND)=DEXP(TMP)
      GINT=TEMP(IND)
      RETURN
      ENTRY GINT0(U)
      IND=IND+1
      GINT0=TEMP(IND)*TEMPP(IND)**J*TEMPS(IND)**K
      RETURN
      END
```

```
C       LISTING 2
C       CALCULATES THE NECESSARY DOUBLE INTEGRALS, OPTIMAL
C       ESTIMATE T(Y), MSE(T(Y)), AND WEIGHTS, FIRST FOR
C       GAUSSIAN CONFIGURATIONS EVALUATED AS GAUSSIAN AND
C       SLASH CONFIGURATIONS, THEN FOR SLASH CONFIGURATIONS
C       EVALUATED AS GAUSSIAN AND SLASH CONFIGURATIONS.
C       CAN BE EXPANDED TO CWG, LOGISTIC, AND CAUCHY.
C       OTHER COMMENTS AS IN LISING 1.
C
C
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 C(100),ANS(12)
        COMMON /AREA1/C,N/AREA4/RST,SST/AREA2/L,K,J/AREA3/V/AREA5/IND
        EXTERNAL FINT,FINTO
        DSEED=13271311.D0
        READ(5,150) N,NSAMP
 150    FORMAT(1I2,1I3)
        DNSAMP=DFLOAT(NSAMP)
        DO 500 LL=1,3,2
        DO 450 LOOP=1,NSAMP
        CALL RANDEV(LL,DSEED)
        CALL SORTS(C,N)
        WRITE(8,410) (C(I),I=1,N)
        NN=(N+1)/2
        NN1=(NN+1)/2
        NN2=N-NN1+1
        R=(C(NN2)+C(NN1))/2.D0
        S=C(NN2)-C(NN1)
        DO 250 I=1,N
 250    C(I)=(C(I)-R)/S
        RST=R
        SST=S
        SUM=0.D0
        DO 350 L=1,3,2
        CONS=CONST(L)
        IF(L.EQ.1) INC=0
        IF(L.EQ.3) INC=4
        K=0
        J=0
        IND=0
        CALL DINT24(FINT,F00)
        K=2
        J=0
        IND=0
        CALL DINT24(FINT0,F20)
        F20=F20/F00
        K=1
        J=1
        IND=0
        CALL DINT24(FINT0,F11)
        F11=F11/F00
        K=0
        J=2
```

```
      IND=0
      CALL DINT24(FINT0,F02)
      F02=F02/F00
      F00=F00*CONS
      PITC=-F11/F20
      PITX=P+S*PITC
      PITMSE=F11*PITC+F02
      ANS(INC+1)=F00
      ANS(INC+2)=F20
      ANS(INC+3)=PITX
      ANS(INC+4)=PITMSE
      SUM=SUM+F00
 350  CONTINUE
      ANS(1)=ANS(1)/SUM
      ANS(5)=ANS(5)/SUM
      WRITE(8,400)(ANS(II),II=1,8)
 400  FORMAT(3(4D16.7/))
 410  FORMAT (7D16.7)
 450  CONTINUE
 500  CONTINUE
      STOP
      END
C
      DOUBLE PRECISION FUNCTION FINT(V)
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL GINT,GINT0
      COMMON /AREA3/VV
      VV=V
      CALL AINT24(GINT,ANS)
      FINT=ANS
      RETURN
      ENTRY FINT0(V)
      VV=V
      CALL AINT24(GINT0,ANS)
      FINT0=ANS
      RETURN
      END
C
      SUBROUTINE RANDEV(LL,DSEED)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 C(100)
      COMMON /AREA1/C,N
      DATA PI/3.14159265357828/
      GO TO (10,20,30,40,50),LL
 10   DO 15 I=1,N
 15   C(I)=GGNQF(DSEED)
      RETURN
 20   DO 25 I=1,N
 25   C(I)=GGNQF(DSEED)
      C(N)=10.D0*C(N)
      RETURN
 30   DO 35 I=1,N
 35   C(I)=GGNQF(DSEED)/GGUBFS(DSEED)
```

```
      RETURN
 40   DO 45 I=1,N
      ARG=GGUBFS(DSEED)
 45   C(I)=DLOG(ARG/(1.D0-ARG))
      RETURN
 50   DO 55 I=1,N
 55   C(I)=DTAN(PI*(GGUBFS(DSEED)-0.5D0))
      RETURN
      END
C
      DOUBLE PRECISION FUNCTION CONST(L)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 C(100)
      COMMON /AREA1/C,N
      DATA PI/3.141592653578280/
      CONST=1.D0
      GO TO (1,1,1,2,2),L
 1    CONST=1.D0/DSQRT(2.D0*PI)
      CONST=CONST**N
      RETURN
 2    CONST=(1.D0/PI)**N
 3    RETURN
      END
C
      DOUBLE PRECISION FUNCTION CINT(U)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 Z(100),TEMPS(576),TEMPP(576),TEMP(576)
      COMMON/AREA1/Z,N/AREA4/PST,SST/AREA2/L,K,J/AREA3/V/AREA5/IND
      IND=IND+1
      DN=DFLOAT(N)
      POW=1.D0/DSQRT(DN)
      TEMPS(IND)=SST*(((1.D0-U)/U)**POW)
      TEMPP(IND)=TEMPS(IND)*POW*DLOG((1.D0-V)/V)+PST
      TEMP(IND)=DN*DLOG(TEMPS(IND))-DLOG(DN*U*V*(1.D0-U)*(1.D0-V))
      SUM1=0.D0
      SUM2=0.D0
      GO TO(10,20,30,40,50),L
 10   DO 15 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
 15   SUM1=SUM1-X*X/2.D0
      GO TO 60
 20   DO 25 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
      X2D2=X*X/2.D0
      EXPON=X2D2*3.99D0
      IF(EXPON .GT. 170.D0) EXPON=170.D0
      SUM1=SUM1-X2D2
 25   SUM2=SUM2+DEXP(EXPON)/(DN*10.D0)
      SUM1=SUM1+DLOG(SUM2)
      GO TO 60
 30   DO 35 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
      X2=X*X
```

```
      EXPON=0.5D0*X2
      IF(EXPON .GT. 170.D0) EXPON=170.D0
      SUM2=1.D0-DEXP(-EXPON)
35    SUM1=SUM1+DLOG(SUM2/X2)
      GO TO 60
40    DO 45 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
      EXPON=X
      IF(DABS(EXPON) .GT. 170.D0) EXPON=DSIGN(170.D0,EXPON)
      SUM2=1.D0+DEXP(EXPON)
45    SUM1=SUM1+X-2.D0*DLOG(SUM2)
      GO TO 60
50    DO 55 I=1,N
      X=TEMPP(IND)+TEMPS(IND)*Z(I)
55    SUM1=SUM1-DLOG(1.D0+X*X)
60    TMP=TEMP(IND)+SUM1
      IF(DABS(TMP) .GT. 170.D0) TMP=DSIGN(170.D0,TMP)
      TEMP(IND)=DEXP(TMP)
      GINT=TEMP(IND)
      RETURN
      ENTRY GINT0(U)
      IND=IND+1
      GINT0=TEMP(IND)*TEMPP(IND)**J*TEMPS(IND)**K
70    FORMAT(D16.7)
      RETURN
      END
```

```
C     LISTING 3
C     CALCULATES BIEFFICIENT (SLASH,GAUSSIAN) ESTIMATE,
C     SHADOW PRICES, AND EXCESS VARIANCES (SEE TUKEY,1981A)
C     FOR 100 SAMPLES EACH OF GAUSSIAN AND SLASH (SAMPLE SIZE 20).
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(2),B(2),V(2)
      DATA EPS,ITL/1.D-04,10/
      DO 650 K=1,2
      W1 =0.D0
      W2 = 0.D0
      PVG = 0.D0
      PVS = 0.D0
      DO 600 I=1,100
C     READ CONFIGURATION WEIGHTS AND PITMAN VARIANCES
      READ(7,11) WC,SH1,WS,SH2
 11   FORMAT(///,D15.7,22X,D16.7,/,D15.7,22X,D16.7)
      PVG = PVG + WC*SH1
      PVS = PVS + WS*SH2
      W1 = W1 + WC
      W2 = W2 + WS
600   CONTINUE
C     PVG (PVS) IS THE WEIGHTED AVERAGE OF PITMAN VARIANCES FOR
C     GAUSSIAN (SLASH).
      IF(K .EQ. 1) GOTO 620
      PVG = (PVG/W1 + PG)/2.D0
      PVS = (PVS/W2 + PS)/2.D0
      GOTO 650
620   PG = PVG/W1
      PS = PVS/W2
650   CONTINUE
      WRITE(6,23) PVG,PVS
 22   FORMAT(4D16.7)
 23   FORMAT(10X,'THE PITMAN VARIANCES ARE: ',2D16.7)
      REWIND 7
      KH = 0
      A(1) = 0.7
      B(1) = 0.3
400   CONTINUE
      REWIND 7
      REWIND 8
      DO 100 I=1,200
C     CALCULATES OPTIMAL T AND RELATIVE EXCESS VARIANCES FOR SPECIFIED
C     SHADOW PRICES (A AND B).
      READ(7,1) X5,X16,WC,SG,TG,WS,SS,TS
  1   FORMAT(63X,D15.7,//15X,D16.7,/D15.7,2D16.7,/D15.7,2D16.7)
      X1 = (X16 - X5)*(X16 - X5)
      WH1 = SG/PVG
      WH2 = SS/PVS
      T = TG*A(1)*WC*WH1 + TS*B(1)*WS*WH2
      T = T/(WC*A(1)*WH1 + WS*B(1)*WH2)
      AMSG = (TG-T)*(TG-T)*WH1/X1
      AMSS = (TS-T)*(TS-T) * WH2/X1
      WRITE(8,2) T,WC,AMSG,WS,AMSS
```

```
2     FORMAT(5D16.9)
100   CONTINUE
      REWIND 8
      DO 250 K=1,2
      VG = 0.
      VS = 0.
      WH1 =0.
      WH2 = 0.
C     CALCULATES WEIGHTED AVERAGE OF RELATIVE EXCESS VARIANCES FOR T
      DO 200 I =1,100
      READ(8,2) T,WG,AMSG,WS,AMSS
      VG = VG + WG*AMSG
      VS = VS + WS*AMSS
      WH1 = WH1 + WG
      WH2 = WH2 + WS
200   CONTINUE
      IF(K .EQ. 1) GOTO 220
      VG = (VG/WH1 +VVG)/2.
      VS = (VS/WH2 + VVS)/2.
      GOTO 250
220   VVG = VG/WH1
      VVS = VS/WH2
250   CONTINUE
      KH = KH+1
      V(1) = VS - VG
      WRITE(6,5) VG,VS
5     FORMAT(10X,'THE EXCESS VARIANCES ARE: ',2D16.7)
C     ITERATES, CHANGING SHADOW PRICES UNTIL (VG-VS)<.0001
C     OR 10 ITERATIONS
      IF(DABS(VG-VS) .LE. EPS  .OR.  KH .GE. ITL) GOTO 500
      IF(KH .EQ. 1) GOTO 300
      AH = A(1) - V(1)*(A(1)-A(2))/(V(1)-V(2))
      IF(AH .GE. 1.) AH= 1.
      A(2) = A(1)
      B(2) = B(1)
      A(1) = AH
      B(1) = 1. - AH
      V(2) = V(1)
      GOTO 400
300   A(2) = A(1)
      B(2) = B(1)
      V(2) = V(1)
      IF(VG .EQ. DMAX1(VG,VS)) B(1) = VS/VG*B(1)
      A(1) = 1 - B(1)
      IF(VS .EQ. DMAX1(VG,VS)) A(1) = VG/VS*A(1)
      B(1) = 1-A(1)
      GOTO 400
500   CONTINUE
      WRITE(6,3) A(1),B(1)
3     FORMAT(10X,'THE SHADOW PRICES ARE: ',2D11.4)
      WRITE(6,4) VG,VS
4     FORMAT(10X,'EXCESS VAR ARE: GAUSS: ',D16.7,' SLASH: ',D16.7)
      END
```

```
C      LISTING 4
C
       SUBROUTINE SORT8(V,N)
       IMPLICIT REAL*8 (A-H,O-Z)
       REAL*8 V(N)
C****************************************
C SHELL SORT ALGORITHM CACM JULY 1964
C****************************************
       I=1
 1     I=I+I
       IF(I .LE. N) GO TO 1
       M=I-1
 2     M=M/2
       IF(M .EQ. 0) RETURN
       K=N-M
       DO 4 J=1,K
       L=J
 5     IF(L .LT. 1) GO TO 4
       IF(V(L+M) .GE. V(L)) GO TO 4
       X=V(L+M)
       V(L+M)=V(L)
       V(L)=X
       L=L-M
       GO TO 5
 4     CONTINUE
       GO TO 2
       END
```

```
C     LISTING 5
C
      SUBROUTINE AINT24(FCT,Y)
      DOUBLE PRECISION Y,A,C,FCT
      DATA A/0.5D0/
C
      C=.49759360999851068D0
      Y=.61706148999935998D-2*(FCT(A+C)+FCT(A-C))
      C=.48736427798565475D0
      Y=Y+.14265694314466832D-1*(FCT(A+C)+FCT(A-C))
      C=.46913727600136638D0
      Y=Y+.22138719408709903D-1*(FCT(A+C)+FCT(A-C))
      C=.44320776350220052D0
      Y=Y+.29649292457718390D-1*(FCT(A+C)+FCT(A-C))
      C=.41000099298695146D0
      Y=Y+.36673240705540153D-1*(FCT(A+C)+FCT(A-C))
      C=.37003104789271718D0
      Y=Y+.43095080765976638D-1*(FCT(A+C)+FCT(A-C))
      C=.32404682596848778D0
      Y=Y+.48809326052056944D-1*(FCT(A+C)+FCT(A-C))
      C=.27271073569441977D0
      Y=Y+.53722135057982817D-1*(FCT(A+C)+FCT(A-C))
      C=.21689675381302257D0
      Y=Y+.57752834026862801D-1*(FCT(A+C)+FCT(A-C))
      C=.15752133984808169D0
      Y=Y+.60835236463901696D-1*(FCT(A+C)+FCT(A-C))
      C=.95559433736808150D-1
      Y=Y+.62918728173414148D-1*(FCT(A+C)+FCT(A-C))
      C=.32028446431302813D-1
      Y=Y+.63969097673376078D-1*(FCT(A+C)+FCT(A-C))
      RETURN
      END
```

DAT

ILM

Figure 8: Powers of 10 for the values of g(u,v,c) at which the function is evaluated for integration (r* and s* relocated).

```
K=2
J=0
IND=0
CALL DINT24(FINT0,F20)
F20=F20/F00
K=1
J=1
IND=0
```

```
REAL*8 C(100)
COMMON /AREA1/C,N/AREA6/CONT,ADJ
DATA PI/3.14159265357829/
GO TO (10,20,30,40,50),LI
```

```
20      DO 20 I=1,N
        X=TEMPR(IND)+TEMPS(IND)*Z(I)
```